

In the Claims

The status of claims in the case is as follows:

1 1. [Currently amended] A scalable system for providing a
2 web processing tool, comprising:

3 a browser;

4 a plurality of first clustered servers running first
5 mirror image system and application code;

6 a plurality of second clustered servers running second
7 mirror image system and application code, servers
8 within said second clustered servers periodically
9 replicating with each other so as to maintain data
10 consistency between them;

11 a database server;

12 a first network dispatcher for dynamically balancing
13 client workload by redirecting clients to one of said
14 first clustered servers based on current workload of

15 servers within said plurality of first clustered
16 servers;

17 a second network dispatcher responsive said first
18 clustered servers for dynamically balancing client
19 workload by redirecting clients to one of said second
20 clustered servers based on current workload of servers
21 within said plurality of second clustered servers;

22 an application server asynchronously responsive to said
23 second clustered servers for running agents to process
24 application data requests and bridge said data with
25 respect to said database server and other back end
26 servers.

1 2. [Original] The scalable system of claim 1, said first
2 clustered servers being operable for presenting a graphical
3 user interface to the said browser and for caching data on
4 behalf of an end user.

1 3. [Original] The scalable system of claim 1, said first
2 clustered servers being domino.go servers operable for
3 presenting a graphical user interface to said browser and
4 redirecting said client via said second network dispatcher

5 to a second cluster server.

1 4. [Original] The scalable system of claim 1, said web
2 processing tool being a web requisition catalog application.

1 5. [Original] The scalable system of claim 1, said second
2 clustered servers being operable for performing workflow,
3 providing security, and serving as a document repository.

1 6. [Original] The scalable system of claim 5, said second
2 clustered servers being domino network servers.

1 7. [Original] The scalable system of claim 6, said
2 document repository being requisitions stored in domino .nsf
3 files.

1 8. [Original] The scalable system of claim 2, further
2 comprising an external objects dynamic file for storing
3 external objects in one place for dynamic access by said
4 first clustered servers, and for generating said gui.

1 9. [Original] The scalable system of claim 1, said
2 database server being a relational database server.

1 10. [Original] The scalable system of claim 1, said other
2 back end server comprising an enterprise resource planning
3 system, including an accounting application having an
4 accounts payable function.

1 11. [Previously amended] The scalable system of claim 3,
2 further comprising a configuration file of proxy statements
3 for mapping user requests to said second cluster.

1 12. [Currently amended] A method for generating on-line
2 procurement requisitions, comprising the steps of:

3 receiving a client request;

4 dynamically balancing client workload among servers by
5 directing said request to a first server within a first
6 cluster of virtual servers based on current server
7 workload , each server in said first cluster running
8 first same application and system code;

9 operating said first server to determine the mapping of
10 said client request and the function required;

11 responsive to a database access function, dynamically
12 balancing client workload among servers directing said
13 client request to a second server within a second
14 cluster of virtual servers based on current server
15 workload, each server in said second cluster running
16 second same application and system code, servers within
17 said second clustered servers periodically replicating
18 with each other so as to maintain data consistency
19 between them; and

20 operating said second server to direct said client
21 request to an application server where all data is
22 replicated and where bridges and agents execute with
23 respect to data in said database.

1 13. [Original] The method of claim 12, further comprising
2 the steps of:

3 synchronizing all virtual servers within said first
4 cluster; and

5 synchronizing all virtual servers within said second
6 cluster.

1 14. [Original] The method of claim 13, further comprising
2 the steps of:

3 replicating application data to a back-end relational
4 database server; and

5 replicating application data to a back-end enterprise
6 resource planning system including an accounting
7 application having an accounts payable function.

1 15. [Previously amended] A program storage device readable
2 by a machine, tangibly embodying a program of instructions
3 executable by a machine to perform method steps for
4 processing a client request with respect to a database, said
5 method steps comprising:

6 receiving a client request;

7 dynamically balancing server workload by directing said
8 request to a first server within a first cluster of
9 virtual servers based on current server usage, each
10 server within said first cluster executing first same
11 code;

12 operating said first server to determine the mapping of
13 said client request and the function required;

14 responsive to a database access function, dynamically
15 balancing server workload by directing said client
16 request to a second server within a second cluster of
17 virtual servers based on current server usage, each
18 server within said second cluster executing second same
19 code, servers within said second cluster periodically
20 replicating with each other so as to maintain data
21 consistency between them; and

22 operating said second server to direct said client
23 request to an application server where all data is
24 replicated and where bridges and agents execute with
25 respect to data in said database.

1 16. [Previously amended] A computer program product or
2 computer program element for:

3 receiving a client request;

4 dynamically balancing server workload by directing said
5 request to a first server within a first cluster of

6 virtual servers based on current server usage, each
7 server within said first cluster executing first same
8 code;

9 operating said first server to determine the mapping of
10 said client request and the function required;

11 responsive to a database access function, dynamically
12 balancing server workload by directing said client
13 request to a second server within a second cluster of
14 virtual servers based on current server usage, each
15 server within said second cluster executing second same
16 code, servers within said second cluster periodically
17 replicating with each other so as to maintain data
18 consistency between them; and

19 operating said second server to direct said client
20 request to an application server where all data is
21 replicated and where bridges and agents execute with
22 respect to data in said database.

1 17. [Previously submitted] The program storage device of
2 claim 15, said method further comprising the step of
3 operating said first clustered servers for presenting a

4 graphical user interface to the said browser and for caching
5 data on behalf of an end user.

1 18. [Previously submitted] The program storage device of
2 claim 15, said method further comprising:

3 synchronizing all virtual servers within said first
4 cluster; and

5 synchronizing all virtual servers within said second
6 cluster.

1 19. [Previously submitted] The program storage device of
2 claim 18, said method further comprising:

3 replicating application data to a back-end relational
4 database server; and

5 replicating application data to a back-end enterprise
6 resource planning system including an accounting
7 application having an accounts payable function.

1 20. [Previously submitted] The program storage device of
2 claim 15, said method further comprising storing external

3 objects in an external objects dynamic file for dynamic
4 access by said first cluster of servers, and for generating
5 said gui.